CS 6630 – Visualization
Project Proposal
Visualization of Mobile Sensor Network Coverage
Tim Sodergren UID: U0076036

Background and Motivation

Mobile sensor networks are becoming an increasingly pervasive feature of the modern automated world. This has led to an ever-growing body of research in many areas including network coverage problems. Communications networks are arguably the most well-known but there are many others. In robotics, for example, many systems are being set up as a network of machines where the primary features depend on each individual robot or node needs to be aware of its position relative to other nodes to effectively operate together. An extension of this problem would be robot beacon navigation. There is also environmental sensing, radio broadcasting, and surveillance, security and military applications as well. Within these environments, the basic question is the same: given and dynamic and complex environment, how do I determine coverage and optimize the placement and timing of my sensors to maximize sensing coverage.

One particularly interesting area of research along these lines involves the use of topological methods for determining coverage. There is a large body of work from Vin de Silva and others whereby they apply variations of what's called a Rips complex which allows for the determination of network coverage with no sensor coordinate information. It relies only on each node knowing what other notes are within their broadcast radius (de Silva & Ghrist, 2006). Of interest to me is an area that still needs research, namely that of optimizing coverage in an environment where the number of sensors greatly exceeds that which is necessary to cover the domain of interest. Also, I am interested in introducing a stochastic element to their calculations to more accurately represent realistic scenarios. In other words, sensors will not be statically placed, but, rather, will move per some schedule. Also, all sensors will not be available all the time.

Visualization is an essential part of this research. An end user will ultimately need to be able to see exactly where, within a particular geographic area, they have coverage, where they need coverage, how that coverage will change as a function of time, and, in the case where there are redundant nodes, what are the most effective nodes and which nodes can I "turn off" in order to conserve energy.

Project Objectives

While this project will be small part of a much larger body of work, there are several things I would like to accomplish within this scope:

1. I would like to easily visualize, on a map, sensor coverage and coverage gaps. The visualization needs to accurate, with as much detail as possible, but must be easy to see and navigate.
2. I need an ability to add data manually or from a file. More importantly, I need to be able to graphically edit node locations with the goal of optimizing coverage. There will be some calculations in the background, probably in Python as JavaScript is a bit limited in this case. For this project, however, that calculation will only be a measure of coverage; I need graphical tools to help me "fill in the gaps".

3. The design needs to be scalable so that I can evaluate any geographic area within a reasonable size limit. This scalability applies both to the physical dimensions of the study area as well as the sensor node data.

Data

Since this project will ultimately be a part of a larger project for which no data have yet been collected, there will be no data gathering. I plan to generate a reasonable set of synthetic data consistent with the real-world problem. Indeed, node location generation will need to be a key component of this project. I would also like to have an option whereby the interface allows the user to generate a set of randomly distributed nodes to work with.

Data Processing

Given the previous statement, there won't be any data processing necessary, however, I will need to establish an easily understandable data standard so that this tool can be applied to any generic sensor network. I will implement an interactive data loading tool that will facilitate this and ensure that the data are consistent before being integrated in the visualization.

Visualization Design

The main component of the visualization, and one commonality between the attached sketches, is, of course, a map background. This is fundamental to understanding network coverage in spatial dimensions. For this project, I will use the University campus as the primary study area. The map itself should have the usual map features an average user would be accustomed to including zoom and pan operations. It should also allow for displaying or hiding streets, buildings, etc. and allow for satellite or standard map backgrounds. Each node will be displayed in its geographic location. There needs to be a time slider as node location and availability will be dynamic. Users should also be able to "play" the coverage map as a movie with an ability to set the playback speed. Data will be displayed in a separate tab. This is where the options for editing and loading will reside. Additionally, there will be a third tab that allows the user to visualize a single nodes position on the map as a function of time. For each node, when a user clicks, it will be highlighted, and additional information will be displayed to the right of the main map. This will include the node label, coordinates, some metric of the nodes importance relative to overall coverage, what nodes it is connected to (also highlighted) and its schedule. There will be a context menu that will allow different options upon right clicking such as editing or moving the point.

To the right of the main map/tabbed display will be a node information window. Clicking on any node will populate this window with all the relevant information for that node including its label, position, connected nodes (i.e., those that are withing the nodes coverage radius) and graphical information highlighting the nodes relative importance to coverage.

Graphical operations will be grouped to the right of the map, including the ability to insert, delete, or move nodes.

Must-have features

In order to effectively allow a user to determine network coverage, this visualization will need, at a minimum, the following features:

1. Background map. This is necessary to provide a frame of reference and should include at least streets and intersections.
2. Nodes locations and graphical representation of covered and non-covered areas.
3. Ability to add nodes either manually or graphically.
4. Ability to move nodes graphically by selecting the node and clicking on a new area of the map.
5. Coverage vs. time. A user needs to be able to see the coverage map for any time and should have the ability to edit the schedule or any node.

Optional features

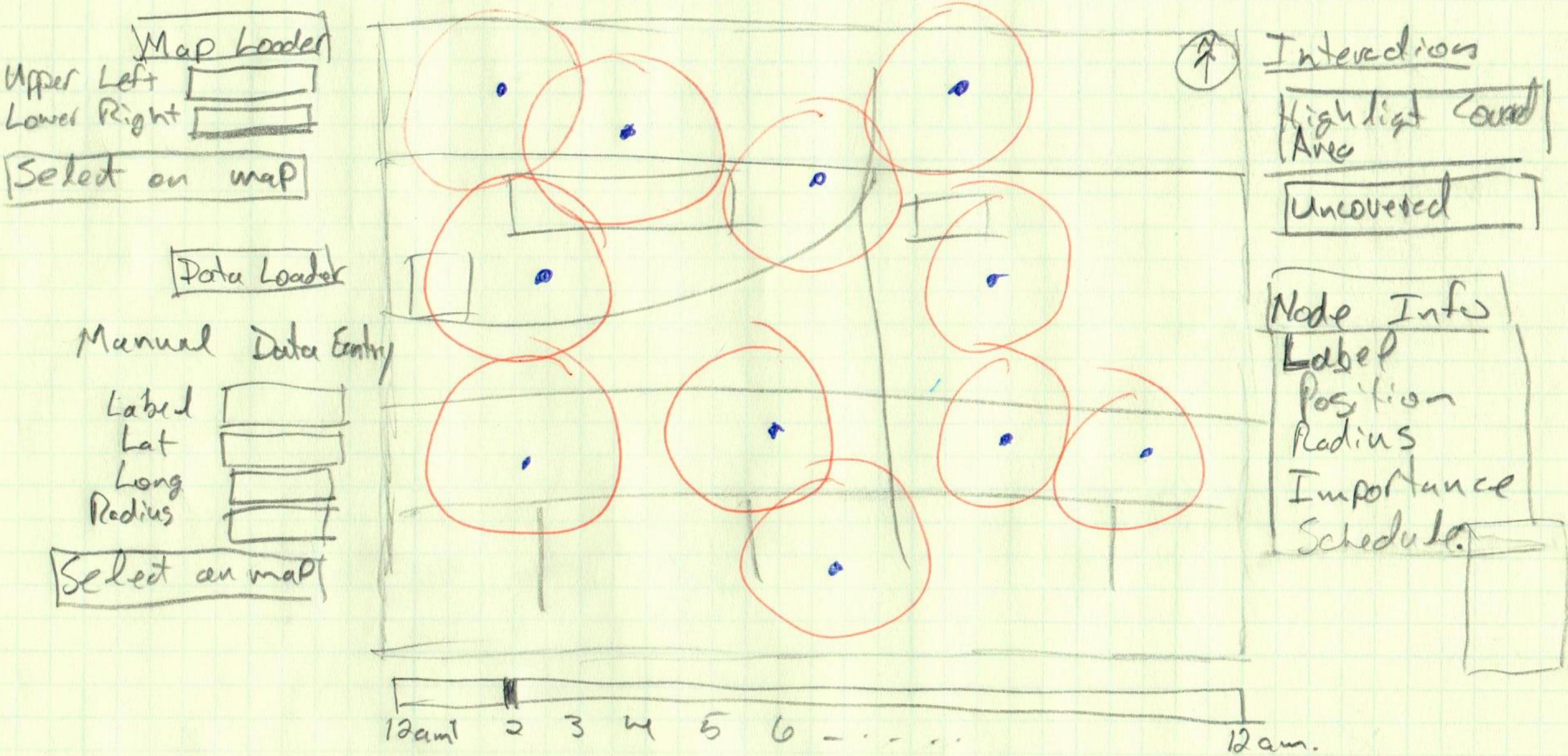
The following features would enhance the overall usability of the design but are not critical for its operation:

1. Additional map options. It would be useful to have the ability to add satellite imagery, topography, and buildings to the view as these would include information that could impact node coverage in a way that cannot be visualized in a simple planar representation.
2. Drag-and-drop nodes. Rather than statically changing node location by clicking on the map or editing in the data spreadsheet, it would be useful to be able to drag a node to a new location. This motion could be linked to the node information window so that the data are updated dynamically as the node is moved around the map, allowing for optimal placement.
3. Aggregate operations. The ability to analyze, move, or edit multiple nodes at once would make the interface easier to use as opposed to having to look at one node at a time. This feature could also provide for some additional insight by allowing a user to see how certain nodes interact with each other.

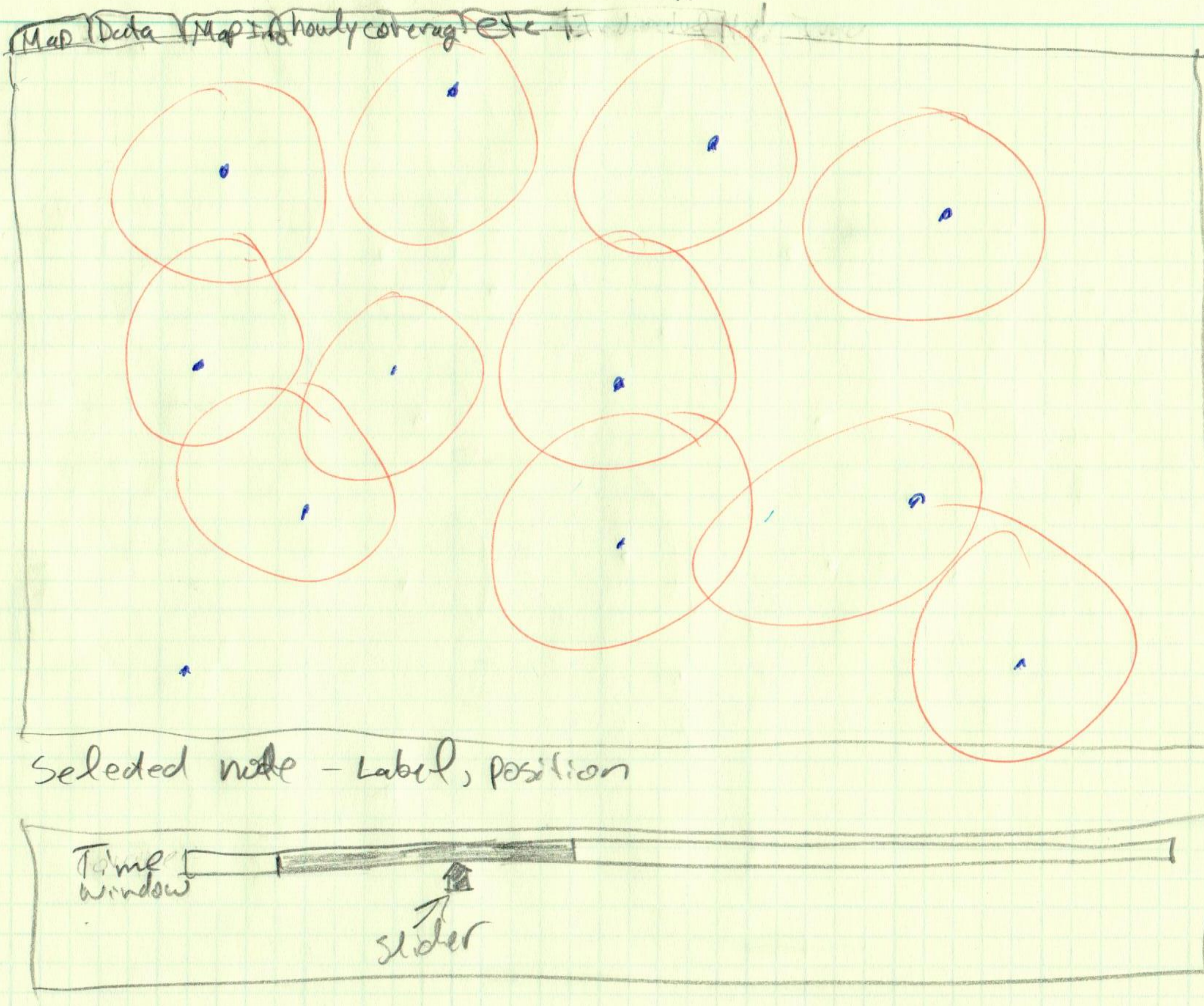
Project Schedule

- 4-Nov: Synthetic data set generated, data import code complete.
- 11-Nov: Static design prototype in place which should include:
    - Basic map view.
    - Node location and coverage areas.
    - Node information window in place and at least showing partial node information for each node clicked.
- 18-Nov: Coverage calculation metric via Python implemented, node editing capability.
- 25-Nov: Remaining features including movie capability
- 25-Nov – 30-Nov: Beta testing, debugging.
- 1-Dec – 2 Dec: Finalize and publish design.

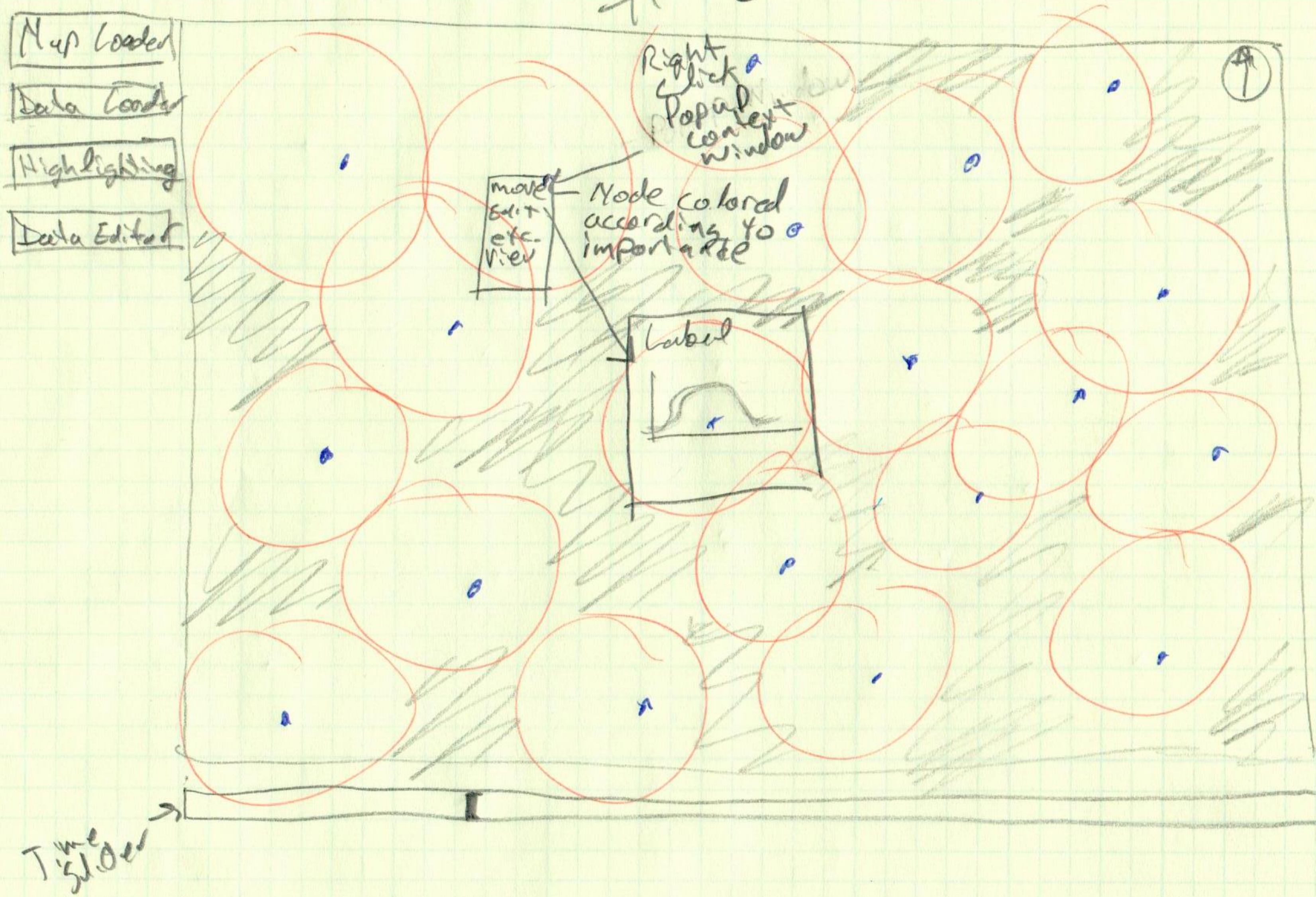
#1


Map Loader
Upper Left
Lower Right
Select on map
Data Loader
Manual Data Entry
Label
Lat
Long
Radius
Select on map
Interaction
Highlight Covered Area
Uncovered
Node Info
Label
Position
Radius
Importance
Schedule
12am 2 3 4 5 6 . . . .
12 am.

#2


Map
Data
Map Info
hourly coverage
etc.
Selected node - Label, position
Time window
slider

#3


Map Loader
Data Loader
Highlighting
Data Editor
Right click Popup context window
move edit etc. view
Node colored according to importance
Label
Time Slider

Final Proposed Vis